

能量高效的无线传感器网络空间范围查询处理算法

刘 亮 秦小麟 郑桂能 李博涵

(南京航空航天大学信息科学与技术学院 南京 210016)

摘 要 现有无线传感器网络空间范围查询处理算法能量消耗大且当节点失效时查询处理过程易被中断,无法返回查询结果.文中提出了一种能量高效的算法 ESA(Energy-efficient Spatial window query processing Algorithm).它将查询区域划分为若干个网格,每个网格中有一个簇头节点负责收集所在网格中其它节点的感知数据,对这些数据进行聚集运算得到部分查询结果,并将其发送至下一个网格的簇头节点,直至收集到查询区域内所有节点的感知数据,以生成最终的查询结果.ESA算法只需查询区域内的节点发送一次数据消息,减少了发送的数据消息数目.根据ESA的能耗公式给出了两种网格划分和簇头节点选择算法,以降低算法分发查询消息消耗的能量.设计了一种利用节点冗余保证查询处理过程鲁棒性的算法,避免了ESA因节点失效而中断.提出了两种感知数据收集算法,以减少簇头节点收集其邻居节点感知数据的能耗.最后,对ESA算法和现有的IWQE算法(Itinerary-based Window Query Execution)的性能进行了系统的理论和实验分析.分析结果表明:在绝大多数情况下,ESA算法在能量消耗、查询成功率和查询结果质量方面优于IWQE算法.

关键词 物联网;无线传感器网络;查询处理;空间范围查询;能量高效;容忍节点失效

中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2011.00763

An Energy-Efficient Spatial Window Query Processing Algorithm in Wireless Sensor Networks

LIU Liang QIN Xiao-Lin ZHENG Gui-Neng LI Bo-Han

(College of Information Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016)

Abstract The energy consumption of existing spatial window query processing algorithms in wireless sensor networks is fairly high. When some sensor nodes fail, the query process of these algorithms is very likely to be interrupted and unable to return query result. An energy-efficient spatial window query processing algorithm called ESA is proposed in this paper. It divides the query region into several grids. Each grid has a cluster node which collects the sensory data in it, aggregates the data to derive partial query result and sends it to the cluster node in the next grid. The above process is repeated until all nodes within the query region are traversed in order to generate the final query result. ESA only requires each node within the query region send data message once, which reduces the data messages. The authors propose two grid dividing and cluster node selection algorithms according to the ESA's energy consumption formula to reduce the energy consumption of distributing the query messages. Then, the authors design a query processing recovery algorithm using node redundancy, which avoids the interruption of ESA due to node failures, and two data collection algorithms to reduce the energy consumption during the process

收稿日期:2011-01-10;最终修改稿收到日期:2011-04-01.本课题得到国家自然科学基金(60673127)、国家“八六三”高技术研究发展计划项目基金(2007AA01Z404)、江苏省支撑计划项目(BE2008135)、高等学校博士学科点专项科研基金(20103218110017)、工信部电子信息产业发展基金、中国博士后科学基金(20100481133)和江苏省博士后基金(1001005B)资助.刘 亮,男,1985年生,博士研究生,主要研究方向为分布式环境数据管理. E-mail: liangliu@nuaa.edu.cn. 秦小麟,男,1953年生,教授,博士生导师,主要研究领域为安全数据库、时空数据库、分布式环境数据管理与容灾等. 郑桂能,男,1988年生,硕士研究生,主要研究方向为传感器网络数据管理. 李博涵,男,1979年生,博士,主要研究方向为传感器网络数据管理.

where the cluster nodes collect the sensory data of its neighbors. Finally, the performance of ESA and IWQE (itinerary-based window query execution) is analyzed systematically. Analytical and experimental results show that in most cases ESA outperforms IWQE in terms of energy consumption, query success rate and query result quality.

Keywords Internet of Things; wireless sensor networks; query processing; spatial window query; energy efficiency; node failure tolerance

1 引 言

无线传感器网络作为物联网的重要组成部分,在灾害搜救、环境监测、医疗卫生、智能家居等领域有着广阔的应用前景,是当前信息技术领域的一个研究热点.在许多传感器网络应用中,用户经常提交空间范围查询^[1]以获得网络覆盖范围内某局部区域的信息,如当前时刻某区域核辐射最大值、受困人员数目、海洋某区域的平均温度等.考虑到传感器节点的能量通常由电池供应十分有限,且受传感器网络规模、工作环境等因素的制约,难以为节点补充能量.因此,能量高效的空间范围查询处理是目前迫切需要解决的问题.

针对传感器网络环境下的空间范围查询,现有基于树的算法^[2-12]依赖于预先构造的树形网络拓扑.由于受周围环境和节点移动的影响,传感器网络拓扑结构频繁发生变化,导致维护树形网络拓扑的代价太大.现有基于路线的算法 IWQE^[1] (Itinerary-based Window Query Execution)沿一条或多条根据网络拓扑动态生成的路线收集查询区域内节点的感知数据,有效地降低了网络拓扑变化对查询处理的影响,但该算法还存在如下问题:(1)能量消耗较大;(2)节点失效时查询结果全部丢失;(3)当查询区域内存在不包含节点的子区域时,无法得到正确的查询结果.

针对这些问题,本文提出了一种能量高效的传感器网络空间范围查询处理算法 ESA(Energy-efficient Spatial window query processing Algorithm).在查询处理过程中,ESA 根据网络拓扑将查询区域划分为一系列网格,在每个网格中挑选出一个簇头节点用于收集网格中其它节点的感知数据,对收集到的数据集合进行聚集运算(如求最大值、求和等),然后将聚集后生成的部分查询结果发送至下一个网格的簇头节点,如此继续,直至到达最后一个网格,该网格的簇头节点生成最终的查询结果,将其返回

至基站.本文提出了两种网格划分和簇头节点选择算法,以减少分发查询消息的能耗;提出了两种分别用于稳定网络和不稳定网络的感知数据收集算法,以减少簇头节点收集其邻居节点感知数据的能耗.为了降低算法因节点失效而中断的概率,本文设计了一种利用节点冗余恢复查询处理过程的算法.理论和实验结果表明:在绝大多数情况下,ESA 算法优于现有的 IWQE 算法.

相比现有的算法,ESA 算法具有如下优点:查询区域内的所有节点只需发送一次数据消息,减少了发送的数据消息数目.当节点失效时,ESA 算法利用备份节点恢复查询处理过程,提高了算法的鲁棒性.当查询区域中的某个子区域不存在节点时,ESA 利用位置路由协议绕过该子区域,消除了空洞区域对查询处理的影响.ESA 仅要求节点维护其邻居节点位置信息,根据该信息动态地进行网格划分和选择网格的簇头节点,避免了网络拓扑变化对查询处理的影响.

本文第 2 节介绍相关工作和背景知识,并分析现有算法存在的问题;第 3 节首先通过一个示例阐述本文提出的算法 ESA 的思想和优势,然后给出 ESA 的能量消耗公式,并对 ESA 中的关键问题——网格大小设置、网格簇头节点的选择策略等进行研究,并对 ESA 算法和现有的 IWQE 算法的能耗进行理论分析;第 4 节通过实验对 ESA 和 IWQE 算法进行比较;第 5 节总结全文并给出未来的研究方向.

2 相关工作

现有空间范围查询处理算法可以分为两类.

(1) 基于树的算法.文献^[2]提出了 TAG(Tiny Aggregation service for ad-hoc sensor networks)算法.它通过消息泛洪协议将查询消息发送到网络中的所有节点,组织这些节点成一棵以基站为根的树.满足查询条件的节点将感知数据发送至其父节点,

父节点对自己和孩子节点发送的感知数据进行聚集, 将聚集运算后的结果发送到上一层节点, 如此继续, 直至到达基站, 生成最终的查询结果. 文献[3-5]将空间索引 R 树^[13]的思想应用于 TAG 中, 每个节点维护其孩子节点的最小边界矩形. 节点收到查询消息后, 仅将查询消息发送到最小边界矩形与查询区域相交的子节点, 减少了分发查询消息的能耗. 文献[6]提出了 TAG 的改进算法 STWinFlood (Spatio-Temporal Window Flood algorithm). 它包含 3 个阶段: ① 基站利用位置路由协议^[14]将查询消息发送到查询区域内的查询协调节点, 即距离查询区域中心最近的节点; ② 查询协调节点将查询消息泛洪至查询区域内的所有节点, 这些节点收到查询消息后, 将感知数据按查询消息泛洪过程中形成的自己到查询协调节点的最短路径返回至查询协调节点; ③ 查询协调节点按第一阶段发现的路由将查询结果返回至基站. 相比 TAG, STWinFlood 仅将查询消息发送到满足查询条件的节点, 减少了能量的消耗. 文献[7-12]研究了连续近似聚集查询, 利用节点感知数据具有时间相关性减少了相似感知数据的发送. 文献[15-16]利用采样技术减少参与查询处理的节点数目, 通过设置合理的采样概率保证查询结果质量满足精度要求的同时尽量降低能耗. 文献[17]针对当查询区域内节点非均匀分布时, 现有算法的查询结果不准确, 提出了三种感知数据加权算法以提高查询结果的正确性.

(2) 基于路线的算法. 由于传感器网络拓扑结构频繁发生变化, 为了保证查询结果的正确性, 基于树的算法需要维护树形网络拓扑, 这样带来了大量的能量消耗. 针对该问题, 文献[1]提出了一种基于路线的数据收集算法 (Itinerary-based data Collection algorithm, 用 IC 表示). 它沿一条根据网络拓扑动态生成的路线收集查询区域内节点的数据. 路线上的节点 (查询节点) 首先将查询消息 (包含查询区域、路线、聚集函数和查询字段等信息) 广播至其邻居节点 (数据节点). 紧接着, 数据节点收到查询消息后将其感知数据发送给查询节点, 最后查询节点对本地的感知数据、从上一个查询节点接收到的部分查询结果和数据节点的感知数据进行聚集运算, 生成新的部分查询结果, 并将其发送给下一个查询节点, 如此继续直至收集到查询区域内所有节点的感知数据.

以图 1 中的网络为例, 图中两个节点之间的边表示它们能够互相通信. 以 S_1 为起点, 采用 IC 算法

计算区域 ABCD 最大温度的过程如图 2 所示. 节点 S_1 先将查询消息广播至其邻居节点, 当前时刻 S_1 的邻居节点 $\{a, b, c, d, S_2\}$ 收到查询消息后, 分别将其监测的温度数据发送给节点 S_1 . S_1 收到邻居节点发送的温度数据后, 计算出节点集合 $\{a, b, c, d, S_1, S_2\}$ 的最大温度 $\max\{a, b, c, d, S_1, S_2\}$. 然后节点 S_1 根据其当前时刻的邻居节点信息选择在查询路线方向前进距离最大的节点 S_2 作为下一个查询节点, 将 $\max\{a, b, c, d, S_1, S_2\}$ 的值发送至 S_2 . 值得注意的是, 若此时有新的节点移动到节点 S_1 的通信半径内且该节点在查询路线方向上的前进距离最大, 则 S_1 选择该节点作为下一个查询节点. S_2 收到消息后, 广播查询消息至其邻居节点, S_2 的邻居节点 $\{a, b, c, d, e, f, S_1, S_3\}$ 中未发送感知数据的节点 $\{e, f, S_3\}$ 将感知数据发送至 S_2 , S_2 利用 S_1 发送的包含 $\max\{a, b, c, d, S_1, S_2\}$ 的消息、邻居节点发送的感知数据可以计算出 $\max\{a, b, c, d, e, f, S_1, S_2, S_3\}$, 并将其发送至下一个查询节点. 如此继续, 直至到达节点 S_4 , S_4 计算出区域 ABCD 的最大温度.

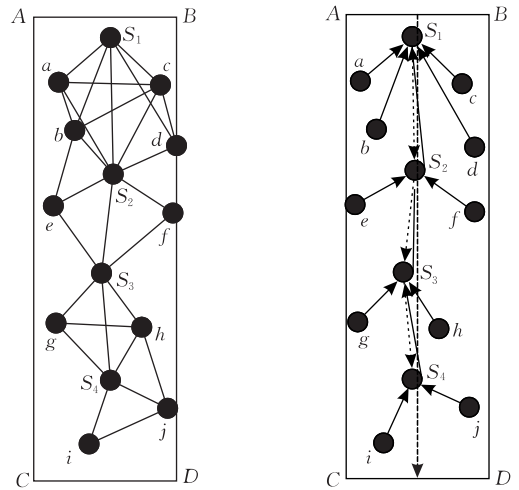


图 1 示例网络 图 2 基于路线的数据收集算法

对于空间范围查询, 文献[1]提出了基于 IC 的 IWQE 算法 (查询路线如图 3 所示). 它首先利用位置路由协议^[14]将查询消息发送至查询区域内的一个节点, 然后该节点利用 IC 算法沿图 3 所示的路线收集感知数据并对查询区域内节点的感知数据进行聚集运算, 计算出最终的查询结果, 最后利用位置路由协议将其返回至基站. 为了保证能收集到查询区域内所有节点的数据, 文献[1]证明查询路线的宽度不能超过节点通信半径的 $\sqrt{3}/2$ 倍. 另外, 文献[18-20]利用 IC 算法处理 K 近邻查询, 即获得距离某位置最近的 K 个节点的感知数据.

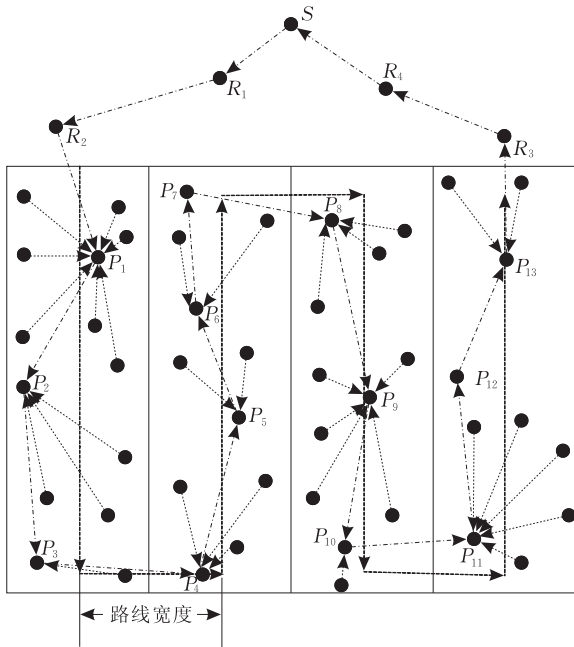


图 3 IWQE 算法

基于路线的算法根据实时的网络拓扑信息动态地形成查询路线,因而相比基于树的算法,能有效地消除网络拓扑变化对查询处理的影响.但它还存在如下问题:(1)能量消耗较大.理想情况下,查询区域中的每个节点只需发送一次数据消息.但若采用基于路线的算法,每个数据节点需要发送一次感知数据消息,而所有查询节点(除第一个查询节点外)需要发送一次感知数据消息和一次部分查询结果消息,即发送本地的感知数据至上一个查询节点和发送聚集后的部分查询结果至下一个查询节点;(2)当查询节点失效时查询结果全部丢失.如图 2 所示,在查询处理过程中,当查询节点(如 S_2 、 S_3 等)失效时,则查询结果丢失且查询处理过程无法继续进行,无法返回查询结果;(3)无法保证查询线路的连通性.在图 2 中,若节点 e 、 f 、 S_3 不存在,则节点 S_2 收集完其邻居节点的感知数据后,查询处理过程被迫中断,无法收集到节点 g 、 h 、 i 、 j 、 S_4 的感知数据.

3 能量高效的传感器网络空间查询处理算法

3.1 假设与符号定义

本文采用文献[1, 18-20]相同的假设.所有传感器节点均维护其一跳邻居节点的位置信息.节点传输半径相等且为 R .平均每个节点有 n_n 个邻居节点.节点发送和接收 1 字节数据消耗的能量分别为 E_t 和 E_r ,广播一个 1 字节数据时,广播节点和接收节

点消耗的能量之和为 E .易知 $E = E_t + n_n \times E_r$.由于无线通信的广播特性,节点单播 1 字节数据时网络消耗的总能量也为 E .点 a 和 b 之间的距离记为 $fd(a, b)$.空间范围查询用 $SWQ(sw)$ 表示,其中 sw 为一矩形查询区域,其宽度、高度、面积分别记为 w 、 h 和 A . sw 中的节点数为 M .查询消息包的大小为 S_q .感知数据消息、经过聚集运算后的部分查询结果和最终的查询结果消息大小均相等且为 S_a .处理空间范围查询网络消耗的总能量 $E_{total} = E_f + E_c + E_b$,其中, E_f 表示将查询消息发送到查询区域内的一个节点消耗的能量; E_c 表示在查询区域内分发查询和收集感知数据消耗的能量; E_b 表示将最终的查询结果返回基站消耗的能量.算法执行过程中发送的感知数据消息、部分查询结果消息统称为数据消息.

3.2 算法思想

为了避免现有算法的问题,本文提出一种能量高效的空間范围查询处理算法 ESA,如图 4 所示,ESA 算法分为 3 个阶段:

(1) 利用位置路由协议将查询消息发送至查询区域内的一个节点.本文称该节点为查询区域起始节点,如图 4 中的节点 a .

(2) 从查询区域起始节点开始,根据实时的网络状态将查询区域划分为若干个网格.每个网格中有一个簇头节点 c_g 负责收集所在网格 g 中其它节点的数据,根据其邻居节点位置信息计算下一个网格 gn 的大小及下一个网格的簇头节点 c_{gn} ,并将查询消息和经过聚集运算后生成的部分查询结果广播至簇头节点 c_{gn} .网格大小和网格簇头节点满足以下约束条件:

ESA 约束条件:网格 gn 中的所有节点均在簇头节点 c_g 的通信半径内,且网格 gn 中除簇头节点 c_{gn} 之外的所有节点均能与 c_{gn} 通信.

由于无线通信的广播特性, c_{gn} 收到 c_g 发送的部分查询结果和查询消息的同时, gn 中的所有节点均能收到该消息. gn 中的节点收到 c_g 发送的查询消息后,依次广播其感知数据至 c_{gn} . c_{gn} 收到 gn 中节点发送的感知数据后,根据 c_g 发送的部分查询结果、 gn 中节点发送的感知数据和 c_{gn} 本地的感知数据,生成新的部分查询结果,并将该部分查询结果和查询消息广播至下一个网格的簇头节点.如此继续,直到遍历完查询区域内的所有节点,计算出最终的查询结果.

(3) 查询区域最后一个网格的簇头节点利用位置路由协议将第二阶段生成的最终查询结果返回至基站.

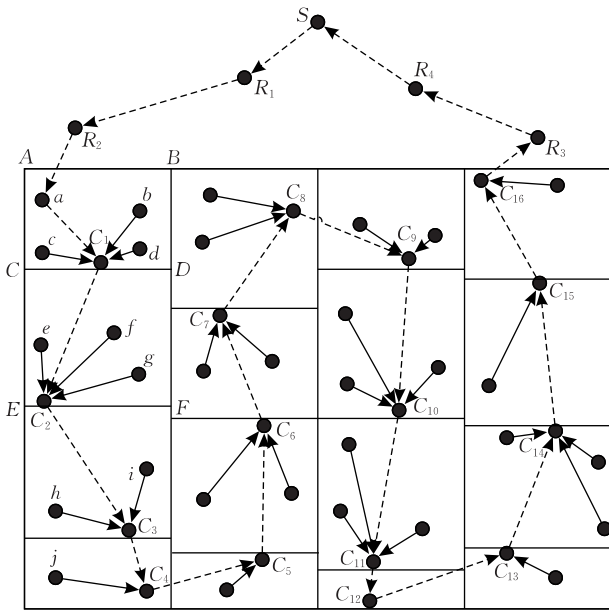


图 4 ESA 算法

对于图 1 中的网络,以 S_1 为起点,采用 ESA 算法计算区域 $ABCD$ (其宽度小于节点间通信半径)最大温度的过程如图 5 所示.由于节点集合 $\{a, b, c, d, S_2\}$ 均在节点 S_1 的通信半径内,且 S_2 在 $\{a, b, c, d\}$ 的通信半径内.节点 S_1 选择其邻居节点 S_2 作为簇头节点收集网格 $ABEF$ 中节点的数据.它先将查询消息和它的温度数据广播至其邻居节点 S_2 ,由于无线通信的广播特性,节点 a, b, c, d 均能接收到该消息.邻居节点 $\{a, b, c, d\}$ 收到查询消息后分别将其感知的温度数据广播给节点 S_2 . S_2 收到邻居节点 $\{a, b, c, d\}$ 和 S_1 发送的温度数据后,计算出到网格 $ABEF$ 为止的部分查询结果,即节点集合 $\{a, b, c, d, S_1, S_2\}$ 的最大温度 $\max\{a, b, c, d, S_1, S_2\}$.由于 $\{e, f, S_3\}$ 均能收到 S_2 发送的消息且 S_3 在 $\{e, f\}$ 的通信半径内,节点 S_2 选择 S_3 收集网格 $EFGH$ 中的数据. S_2 将查询消息和 $\max\{a, b, c, d, S_1, S_2\}$ 的值广播至 S_3 .节点 $\{e, f\}$ 均能收到该消息包.它们收到消息后,将温度数据发送至 S_3 , S_3 利用 S_2 发送的包含 $\max\{a, b, c, d, S_1, S_2\}$ 的消息、 $\{e, f\}$ 发送的温度数据可以计算出到网格 $EFGH$ 为止的部分查询结果 $\max\{a, b, c, d, e, f, S_1, S_2, S_3\}$,并将其广播至下一个簇头节点.如此继续,直至到达节点 i , i 计算出区域 $ABCD$ 的最大温度.

在上述例子中,为了计算区域 $ABCD$ 的最大温度,ESA 算法需要发送 4 个查询消息和 13 个数据消息(包含感知数据消息和部分查询结果消息),而基于路线的算法需要发送 4 个查询消息和 16 个数据消息.可见,ESA 减少了发送的数据消息数目,从

而降低了能量的消耗.本文将在 3.4 节对两种算法的能量消耗进行理论分析.

由于受到节点软硬件故障、节点能量不足、节点休眠等因素的影响,传感器节点失效频繁发生.ESA 算法利用无线通信的广播特性,以避免节点失效对查询处理的影响.如图 5 所示,在 S_2 收集网格 $ABEF$ 的感知数据过程中,由于节点 $\{a, b, c, S_1\}$ 均在彼此的通信半径内,根据无线通信的广播特性,节点 $\{a, b, c, S_1\}$ 均能分别计算出网格 $ABKL$ 的部分查询结果 $\max\{a, b, c, S_1\}$.以节点 a 为例,节点 S_1 、 b 、 c 发送感知数据至节点 S_2 时,节点 a 能监听到这些信息.节点 a 根据这些接收到的感知数据能计算出网格 $ABKL$ 的部分查询结果 $\max\{a, b, c, S_1\}$.当查询处理过程中节点 S_2 失效时,可以从这些节点恢复执行,继续收集区域 $KLCD$ 中的数据.同理,当节点 S_4 失效时,节点 g 和 h 可以作为节点 S_4 的备份节点继续查询处理过程.可见,相比 IWQE 算法,ESA 算法利用节点冗余恢复查询处理过程,降低了算法因节点失效而中断的概率.

另外,当查询区域中的某子区域不存在节点时,ESA 算法利用位置路由协议绕过该子区域继续执行.如图 6 所示, $fd(S_2, G)$ 和 $fd(S_2, K)$ 的大小为节点的通信半径且 $fd(E, G) < fd(F, K)$.由于节点 S_2 在区域 $EFGK$ 中没有邻居节点,区域 $EFGK$ 成为“空洞区域”,查询处理无法继续进行.为了避免该问题,本文采取如下方法: S_2 收集完网格 $ABEF$ 中节点的感知数据后,以网格 $GHIJ$ 的中心为目的位置(其中 $fd(G, J)$ 的大小为节点的通信半径),利用位置路由协议将部分查询结果和查询消息发送至网格 $GHIJ$ 中的节点以绕开“空洞区域”.在图 6 中,

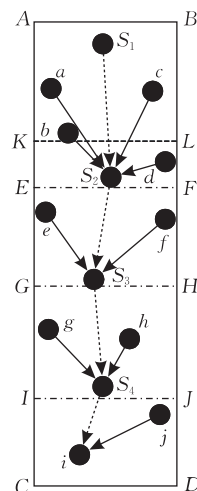
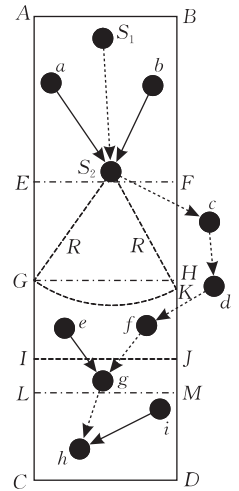
图 5 利用 ESA 算法计算区域 $ABCD$ 的最大温度示意

图 6 绕过不存在节点的区域示意

以网格 $GHIJ$ 的中心为目的位置,部分查询结果和查询消息通过位置路由协议发送至网格 $GHIJ$ 中的节点 f 后,从节点 f 开始继续查询处理过程。

3.3 算法设计

下面首先对 ESA 算法的能耗公式进行推导,然后根据该公式给出降低 ESA 能耗的方法,最后基于该方法优化 ESA 算法的参数设置。

假设 ESA 算法将查询区域划分为 N_g 个网格 $g_i (1 \leq i \leq N_g)$, 在网格 g_i 中的节点集合记为 $ns_i (1 \leq i \leq N_g)$, 对应的簇头节点记为 $c_i (1 \leq i \leq N_g)$. ESA 约束条件可以形式化为

$$\begin{aligned} fd(n, c_i) \leq R, \quad \forall n \in ns_i - c_i, \quad 1 \leq i \leq N_g \\ fd(c_{i-1}, n) \leq R, \quad \forall n \in ns_i, \quad 2 \leq i \leq N_g \end{aligned} \quad (1)$$

由 ESA 算法的执行过程可知,查询区域起始节点需要发送一次查询消息给 c_1 . 所有簇头节点(除最后一个簇头节点外)需将部分查询结果和查询消息发送至其下一个簇头节点. 查询区域内除簇头节点之外的节点均需将感知数据发送至簇头节点. 因此,ESA 算法在查询区域内分发查询和收集感知数据消耗的能量为

$$\begin{aligned} E_c(\text{ESA}) = S_q E + (N_g - 1) S_q E + \\ (N_g - 1) S_a E + (M - N_g) S_a E \end{aligned} \quad (2)$$

化简后,可以得到

$$E_c(\text{ESA}) = N_g S_q E + (M - 1) S_a E \quad (3)$$

可以看出,其能量消耗由两部分组成:(1)发送数据消息带来的能耗,即查询区域内的非簇头节点将感知数据发送至其所在网格的簇头节点;簇头节点将部分查询结果发送至下一个簇头节点.(2)发送查询消息带来的能耗.即查询区域起始节点发送一次查询消息至第一个簇头节点,查询区域内每个簇头节点(最后一个簇头节点除外)发送一次查询消息至其下一个簇头节点。

为了计算精确的查询结果,理论上查询区域内的所有节点(生成最终查询结果的节点除外)均需至少发送一次感知数据消息或部分查询结果消息. 因而根据式(3)可知,ESA 算法发送数据消息的能耗无法降低. 因此,为了减少 ESA 算法的能量消耗,应在满足 ESA 约束条件的前提下尽量降低发送查询消息带来的能耗,即减少簇头节点的数目. 显然,簇头节点的数目与网格的大小、网格簇头节点选择策略有关. 为了求出 N_g 的最优解,需要收集查询区域内节点的拓扑信息,这会带来大量的能量消耗. 另外,即使能获得这些信息,受节点移动、周围环境等因素的影响,网络拓扑结构频繁发生变化,求出的

N_g 最优解变得毫无意义。

基于上述分析,本文采用如下方法:在查询处理过程中,网格簇头节点根据其所有邻居节点当前的位置,动态地选择下一个网格的簇头节点和计算下一个网格的大小. 考虑到 $N_g = A/A_g$, 其中 A_g 表示网格的平均面积,由式(3)可知, A_g 越大,则 E_c (ESA) 越小. 因此,在满足 ESA 约束条件的前提下,使得每个网格 $g_i (1 \leq i \leq N_g)$ 的面积尽可能大,可以增大 A_g , 从而降低算法的能量消耗。

算法 1. GBA.

输入: 簇头节点 c, N_c

输出: c 的下一个网格的高度 h ,

c 的下一个网格的簇头节点 c_{gn}

1. $c_{gn} = \text{NULL}$;
2. IF $N_c.size = 0$
3. $h = 0$; RETURN;
4. END IF
5. //对 N_c 中的所有节点按其到线段 CD 的距离从小到大排序
6. $SN_c = \text{sort}(N_c)$;
7. $c_{gn} = SN_c[1]$;
8. $i = SN_c.size$;
9. WHILE $i > 1$
10. //检查 $SN_c[i]$ 是否能与节点集合
11. // $\{SN_c[1], \dots, SN_c[i-1]\}$ 中任何一个节点通信
12. IF $fc(SN_c[i], \{SN_c[1], \dots, SN_c[i-1]\})$ is true
13. $c_{gn} = SN_c[i]$; BREAK;
14. ELSE
15. $i = i - 1$; CONTINUE;
16. END IF
17. END WHILE
18. $h = \text{dist}(c_{gn}, CD)$.

// h 取值为节点 c_{gn} 到线段 CD 的距离

下面给出两种计算网格大小和网格簇头节点的算法:基于簇头节点前进距离的算法 GBA(Grid Based Algorithm)和基于网格面积的算法 GAA(Grid Area based Algorithm). 不失一般性,以图 7 中的网络为例,以 c 为圆心 R 为半径的圆与查询区域交于点 P 和点 S . 显然,区域 $CDPS$ 中的节点均在网格 $ABCD$ 的簇头节点 c 的通信半径内. 区域 $CDPQ$ 中的节点 d, e, f 均在彼此的通信半径内,节点 d, f, g 均能互相通信. 簇头节点 c 在区域 $CDPQ$ 中的邻居节点集合记为 N_c . 假设函数 $ns(x)$ 返回在区域 x 中节点集合. 函数 $\text{dist}(n, l)$ 返回节点 n 到线段 l 的距离. 对于节点 n 和节点集合 s ,若节点 n 能

与节点集合 s 中的所有节点通信, 则函数 $fc(n, s)$ 返回真, 否则返回假. 簇头节点 c 收集完网格 $ABCD$ 中节点的数据后, 继续收集区域 $CDEF$ 中节点的数据. 簇头节点 c 的下一个网格记为 gn , gn 的簇头节点记为 c_{gn} . c 根据其邻居节点信息计算 c_{gn} 和 gn 的高度 h 的方法如下:

(1) 网格大小和网格簇头节点计算方法

① 基于簇头节点前进距离的算法 GBA

GBA 的核心思想是: 对 N_c 中的所有节点按其到线段 CD 的距离从小到大排序, 排序后的节点列表记为 SN_c . 在 SN_c 中查找节点 $SN_c[i]$ (其中 $1 \leq i \leq |SN_c|$), 满足条件: 节点集合 $\{SN_c[1], \dots, SN_c[i-1]\}$ 中的任意一个节点能与 $SN_c[i]$ 通信. 簇头节点 c 的下一个网格的簇头节点设置为满足上述条件且 i 值最大的节点 x . 簇头节点 c 的下一个网格的高度设置为 x 到线段 CD 的距离. GBA 算法的伪代码详见算法 1. 值得注意的是算法 1 的第 2 行, 若 N_c 为空, 则下一个网格的高度设置为 0. 此时, 区域 $CDPQ$ 成为空洞区域, ESA 算法利用位置路由协议绕过该空洞区域.

如图 7 所示, 显然, 只有网格 $CDGH$ 、 $CDIJ$ 、 $CDKL$ 和 $CDMN$ 有可能成为簇头节点 c 的下一个网格. 在这些网格中, 由于节点 g 不在节点 e 的通信半径内, GBA 算法选择网格 $CDKL$ 作为节点 c 的下一个网格, 节点 f 作为下一个网格的簇头节点.

② 基于网格面积的算法 GAA

GAA 的核心思想是: 对 N_c 中的所有节点 s , 计算 s 作为簇头节点时, 其对应的满足 ESA 约束条件且面积最大的网格 g_s ; 然后在 N_c 中选择使得下一个网格面积最大的节点 x 作为下一个网格的簇头节点, 簇头节点 c 的下一个网格设置为 x 对应的满足 ESA 约束条件且面积最大的网格 g_x . 若 N_c 中存在多个节点使得 gn 面积最大, 则在这些节点中选择距离线段 CD 最远的节点作为 gn 的簇头节点. GAA 算法的伪代码与 GBA 类似, 因而在此不再赘述.

如图 7 所示, $N_c = \{d, e, f, g\}$. 若选择节点 e 作为节点 c 下一个网格的簇头节点, 则其对应的满足 ESA 约束条件且面积最大的网格是 $CDKL$, 而节点 d, f 对应的网格为 $CDMN$, 节点 g 对应的网格为 $CDGH$. 由于节点集合 $\{d, e, f, g\}$ 中使得 gn 面积最大的节点为 d, f 且 $dist(d, CD) > dist(f, CD)$. 算法 GAA 选择网格 $CDMN$ 作为簇头节点 c 的下一

个网格, 节点 f 作为下一个网格的簇头节点.

(2) 网格簇头节点失效时如何恢复执行

如图 8 所示, 节点 c 收集完网格 $ABCD$ 中节点的数据后, 将部分查询结果和查询消息发送至下一个网格 $CDMN$ 的簇头节点 f . 节点 f 收集网格 $CDMN$ 中节点的感知数据后, 生成新的部分查询结果并将其发送至节点 i , 继续查询处理过程. 在上述查询处理过程中, 当网格 $CDMN$ 中的非簇头节点 (如节点 d) 失效时, 则仅丢失了该节点的感知数据, 但不影响查询处理过程的继续执行; 而当簇头节点 f 失效时, 则查询处理过程中断且部分查询结果全部丢失. 本文利用节点冗余以避免该问题, 具体如下:

在网格 $CDMN$ 内可以找到一个网格 rg , 网格 rg 中的任意两个非簇头节点能够互相通信, 称该网格为候选恢复网格. 例如图 8 中节点 d, e, g 均能互相通信, 因而网格 $CDGH$ 、 $CDIJ$ 、 $CDKL$ 均是候选恢复网格. 本文定义所有候选网格中面积最大的网格为恢复网格. 如图 8 中的网格 $CDKL$. 网格 $CDMN$ 中的所有节点均将感知数据发送到簇头节点后, 由于恢复网格中的任意两个非簇头节点可以互相通信, 因此, 恢复网格中的任意一个节点均能分别计算出到所在恢复网格为止、除簇头节点之外的部分查询结果. 例如, 图 8 中的节点 d, e, g 均能分别计算出到网格 $CDKL$ 为止的部分查询结果 $\max\{a, b, c, d, e, g\}$, 即网格 $ABKL$ 的查询结果. 当簇头节点失效时, 恢复网格中的节点能够代替簇头节点继续查询处理过程. 例如, 当图 8 中的簇头节点 f 失效时, 节点 e 能够代替它继续收集区域 $KLEF$ 中节点的数据.

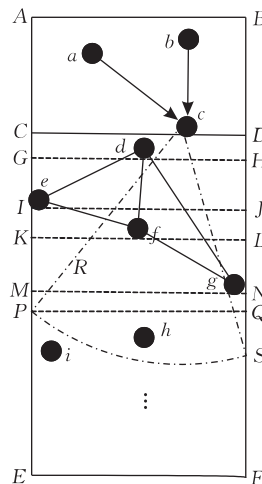


图 7 网格大小和网格簇头节点计算

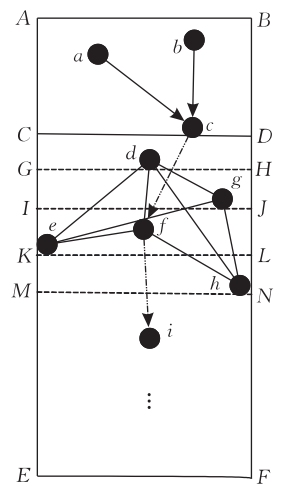


图 8 簇头节点失效时进行的恢复

(3) 感知数据收集算法

簇头节点 c_g 收集所在网格 g 中其它节点的感知数据结束后, 将查询消息和聚集后的部分查询结果发送至下一个网格 g_n 的簇头节点 c_{g_n} . 网格 g_n 中的其它节点均能收到该消息. 若这些节点收到消息后同时发送感知数据至簇头节点 c_{g_n} , 会产生消息碰撞, 造成大量的消息重传. 因此, 需要对其进行调度, 减少消息碰撞的概率, 从而减少能量的消耗. 不失一般性, 以图 9 为例, 簇头节点 c 收集完网格 $ABCD$ 中节点的数据后, 将部分查询结果和查询消息发送至下一个网格 $CDMN$ 的簇头节点 f , 网格 $CDMN$ 中的节点 $\{d, e, f, g, h, i\}$ 均能收到该消息. 这些节点收到查询消息后, 需要将本地的感知数据发送至簇头节点 f . 分两种情况讨论.

情况 1. 簇头节点 f 收集网格 $CDMN$ 中的感知数据时, 网格 g_n 中的节点拓扑稳定. 本文首先提出一种基于距离的算法 DC (Distance-based Data Collection Algorithm), 然后针对该算法的不足, 提出一种改进算法 IDC (Improved Distance-based Data Collection Algorithm).

① 基于距离的感知数据收集算法 DC, 即网格 $CDMN$ 中的节点按其到线段 CD 的距离从小到大排序, 依次发送感知数据消息, 具体过程如下:

1. 对所有在网格 $CDMN$ 中的节点 s , 它收到簇头节点 c 发送的查询消息后, 分别根据自己的邻居节点列表计算出在网格 $CDMN$ 中的邻居节点集合 $ns(CDMN)$;

2. 节点 s 对集合 $ns(CDMN) - f$ 中的节点按其到线段 CD 的距离从小到大排序, 排序后的列表记为 sns , 假设节点 s 在列表 sns 中的位置序号为 i , 其中 $1 \leq i \leq |sns|$, 其中 $|sns|$ 为列表长度;

3. 若 i 为 1, 则节点 s 立即将其感知数据发送至簇头节点 f ; 若 $i > 1$, 则节点 s 等待节点 $sns[i-1]$ 发送感知数据消息结束后, 发送感知数据消息至簇头节点 f .

在 DC 算法中, $\forall i: 1 \leq i < |sns|$, 节点 $sns[i+1]$ 监听节点 $sns[i]$ 发送完感知数据消息后, 立即发送其感知数据, 这样保证消息不发生碰撞的同时减少了感知数据收集过程消耗的时间. 但该算法要求满足条件: 节点 $sns[i+1]$ 在节点 $sns[i]$ 的通信半径内. 上述条件并不恒成立. 如图 9 所示, 若采用 DC 算法收集网格 $CDMN$ 中的感知数据至簇头节点 f , 正确的感知数据发送顺序是 d, e, g, h, i . 但由于节点 e 和 g 、节点 g 和 h 、节点 h 和 i 不能互相通信, 带来以下问题: 节点 i 根据其邻居列表信息可知: 它应在节点 g 之后发送感知数据消息, 但实际上应该在

节点 h 之后; 在感知数据收集过程的起始阶段, 节点 d, g, h 根据其各自的邻居列表信息可知: 没有节点先于它们发送感知数据消息, 因而同时发送感知数据消息至节点 f , 导致消息碰撞. 实际上此时应当只有节点 d 发送感知数据消息.

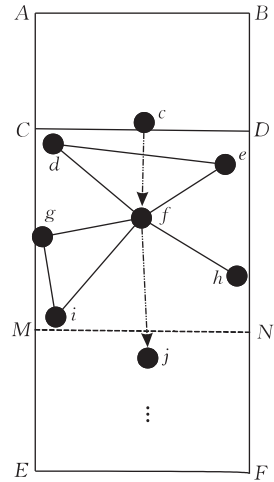


图 9 感知数据收集算法

② 改进的基于距离的感知数据收集算法 IDC

考虑到簇头节点能与所在网格中的任意一个节点通信, 簇头节点根据其邻居节点信息能计算出无消息碰撞的感知数据发送顺序. 下面利用该特性提出一种基于 DC 算法的改进算法 IDC.

IDC 算法用到的符号和定义如下: 网格 $CDMN$ 中的节点集合记为 $ns(CDMN)$. 节点集合 x 中的节点按其到线段 CD 的距离从小到大排序后的有序列表记为 $sns(x)$. $\forall s: s \in ns(CDMN)$, $ng(s)$ 表示节点 s 在网格 $CDMN$ 中、除簇头节点 f 之外的邻居节点. 有序节点列表 x 中在节点 s 之前和之后的节点分别记为 $prev(s, x)$ 和 $next(s, x)$. 将在节点 s 之前和之后发送感知数据消息的节点分别定义为 s 的前驱节点和后继节点. 如图 10(a)~(c) 所示, 节点 s 位于线段 IJ 上. 定义网格 $GHIJ$ 和 $IJKL$ 分别为节点 s 的上网格和下网格, 其中, $fd(I, H) = fd(I, L) = R$. 显然, 可以得出以下结论.

结论 1. 若节点集合 $ng(s)$ 中存在节点在 s 的上网格中 (如图 10(a) 所示), 则等式 $prev(s, sns(ng(s))) = prev(s, sns(ns(CDMN)))$ 成立; 若不存在, 则等式 $prev(s, sns(ng(s))) = prev(s, sns(ns(CDMN)))$ 不一定成立.

结论 2. 若簇头节点 c 在节点 s 的上网格中且 $prev(s, sns(ng(s))) = \text{NULL}$ (如图 10(b) 所示), 则 $prev(s, sns(ng(s))) = prev(s, sns(ns(CDMN))) =$

NULL, 即节点 s 是网格 $CDMN$ 中发送感知数据消息的起始节点。

结论 3. 若 $ng(s)$ 中存在节点在节点 s 的下网格中(如图 10(c)所示), 则等式 $next(s, sns(ng(s))) =$

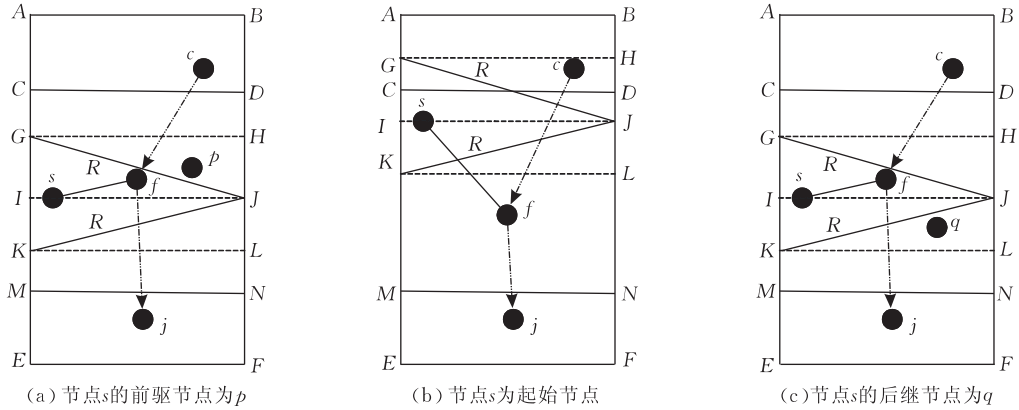


图 10 节点的前驱和后继节点示意图

基于上述分析, IDC 算法流程如下:

对于网格 $CDMN$ 的簇头节点 f , 它收到节点 c 发送的查询消息后, 根据其维护的邻居节点集合计算出网格 $CDMN$ 中发送感知数据消息的起始节点, 并判断簇头节点 c 是否在该起始节点的上网格中; 若否, 则发送 $start$ 消息给该起始节点($start$ 消息用于通知节点开始发送感知数据消息至节点 f), 该起始节点收到 $start$ 消息后, 发送感知数据消息至簇头节点 f .

对于网格 $CDMN$ 的任意一个非簇头节点 s , 它收到节点 c 发送的查询消息后, 进行以下步骤:

1. 根据其维护的邻居节点列表计算出在网格 $CDMN$ 中的邻居节点集合 $ng(s)$, 转到步 2;
2. 节点 s 根据结论 2 判断是否是网格 $CDMN$ 中发送感知数据消息的起始节点, 若是, 则立即将其感知数据发送至簇头节点 f ; 若否, 转到步 3;
3. 节点 s 根据结论 1 判断 $prev(s, sns(ng(s)))$ 是否等于 $prev(s, sns(ns(CDMN)))$:

(a) 若是, 则节点 s 在节点 $prev(s, sns(ng(s)))$ 完成发送感知数据消息后, 立即发送感知数据消息至簇头节点 f ;

(b) 若否, 则等待簇头节点 f 发送的 $start$ 消息. 收到簇头节点 f 发送的 $start$ 消息后, 发送其本地的感知数据至簇头节点 f , 然后转到步 4;

4. 节点 s 根据结论 3 判断 $next(s, sns(ng(s)))$ 是否等于 $next(s, sns(ns(CDMN)))$, 若否, 则发送 $coordinate$ 消息至簇头节点 f . $coordinate$ 消息用于通知簇头节点 f 计算节点 s 的后继节点并将 $start$ 消息发送至该后继节点. f 收到该消息后, 根据其邻居节点列表计算 $next(s, sns(ns(CDMN)))$, 并将 $start$ 消息发送至该节点.

值得注意的是, 在 IDC 算法中, 当网格 $CDMN$

$next(s, sns(ns(CDMN)))$ 成立; 若不存在, 则等式 $next(s, sns(ng(s))) = next(s, sns(ns(CDMN)))$ 不一定成立.

中的节点 s 根据其邻居节点集合计算出的 $next(s, sns(ng(s)))$ 不等于 $next(s, sns(ns(CDMN)))$ 时, 它发送 $coordinate$ 消息至簇头节点 f . 节点 f 根据其邻居节点集合计算 s 的后继节点, 并发送 $start$ 消息至该节点, 从该节点开始继续数据收集过程.

以图 9 为例, 节点 d 根据结论 2 可以判断出它是起始节点. 节点 e 根据结论 1 可知它的前驱节点是 d , 且根据结论 3 可知, 它无法根据自己的邻居节点集合计算出正确的后继节点. 因此, 节点 d 将感知数据发送至簇头节点 f 后, 节点 e 发送 $coordinate$ 消息和感知数据至节点 f . f 收到该消息后, 根据其邻居节点集合, 计算出 e 的后继节点为 g , 然后发送 $start$ 消息至节点 g . 节点 g 收到节点 f 发送的 $start$ 消息后, 继续后面的数据收集过程.

情况 2. 簇头节点 f 收集网格 $CDMN$ 中感知数据时网格 gn 中的节点拓扑频繁变化. 本文提出一种基于角度的感知数据收集算法 AC(Angle-based data Collection algorithm), 即类似于文献[1]基于竞争的算法, 网格 $CDMN$ 中的节点收到簇头节点 c 发送的查询消息后, 各自计算其到簇头节点 f 的角度 a , 并按公式 $max_delay \times (a / (2\pi))$ (其中 max_delay 表示簇头节点完成数据收集花费时间的最大值) 计算其发送感知数据的时间. 该算法的性能取决于 max_delay 的估计值是否准确. 若 max_delay 的估计值比实际值大, 则会导致感知数据收集过程消耗的时间变长; 若估计值比实际值小, 则容易造成消息碰撞问题, 浪费了能量的同时增加了感知数据收集过程消耗的时间. 为了避免这些问题, 本文采用如下

方法:由于网格 gn 中的节点均在簇头节点 c 的通信半径内,因此簇头节点 c 能计算出网格 gn 中的节点数目 $Num(gn)$. 可以得到

$$max_delay = (Num(gn) - 1) \times T \quad (4)$$

其中, T 表示收集一个节点的感知数据需要花费的时间. 簇头节点 c 计算出 max_delay 后, 通过将其附加在查询消息中, 发送至网格 gn 中的节点.

3.4 理论分析

文献[1]没有对 IWQE 算法的能耗进行理论分析, 下面先给出 IWQE 的能耗公式, 然后与本文提出的算法进行比较. 为了便于理论分析, 假设查询处理过程中节点不失效. 记 IWQE 算法的查询节点数目为 N_q , 其路线宽度和 ESA 算法的网格宽度均设置为 w_1 ($w_1 < R$). 由 IWQE 算法的执行过程可知: 每个查询节点需要广播一次查询消息, 查询节点的邻居节点集合中未发送感知数据消息的节点发送感知数据消息至查询节点, 每个查询节点(最后一个查询节点除外)将部分查询结果和查询消息发送到下一个查询节点, 或者当下一个查询节点已经收到了查询节点发送的查询消息时, 则查询节点只需将部分查询结果发送到下一个查询节点. 因此, IWQE 算法在查询区域内分发查询和收集感知数据消耗的能量最大值和最小值分别为

$$E_c^{max}(IWQE) = N_q S_q E + (M - 1) S_a E + (N_q - 1)(S_a + S_q) E \quad (5)$$

$$E_c^{min}(IWQE) = N_q S_q E + (M - 1) S_a E + (N_q - 1)(S_a) E \quad (6)$$

化简后, 可以得到

$$E_c^{max}(IWQE) = (2N_q - 1) S_q E + (M + N_q - 2) S_a E \quad (7)$$

$$E_c^{min}(IWQE) = (N_q - 1) S_q E + (M + N_q - 2) S_a E \quad (8)$$

因此, 由式(7)和(8)可得

$$E_c(IWQE) = (\alpha N_q - 1) S_q E + (M + N_q - 2) S_a E \quad (9)$$

其中, $1 \leq \alpha \leq 2$. 由于在基站、查询区域固定的情况下, $E_f(IWQE) = E_f(ESA)$, $E_b(IWQE) = E_b(ESA)$ 且 E_f 和 E_b 为一常量. 根据式(3)和(9), 可以得出以下结论: IWQE 算法和 ESA 算法的能量消耗均与查询消息大小、感知数据消息大小成正比; 当感知数据消息大小增大时, IWQE 算法消耗能量的增幅大于 ESA 算法.

对于用户提交的快照空间范围查询^[1](例如获得查询区域 sw 当前时刻的平均温度), IWQE 算法

与 ESA 算法消耗的能量之差为

$$E_{total}(IWQE) - E_{total}(ESA) = E_c(IWQE) - E_c(ESA) = (\alpha N_q - N_g - 1) S_q E + (N_q - 1) S_a E \quad (10)$$

其中, $1 \leq \alpha \leq 2$. 化简式(10)后, 可以得到

$$E_{total}(IWQE) - E_{total}(ESA) = [S_q / S_a (\alpha N_q - N_g - 1) + (N_q - 1)] S_a E \quad (11)$$

IWQE 算法的查询节点数目 N_q 为

$$N_q \approx \left\lceil \frac{w}{w_1} \right\rceil \frac{h}{Adv_{IWQE}} \quad (12)$$

其中, Adv_{IWQE} 表示 IWQE 算法中相邻查询节点在查询路线方向上的距离的平均值. ESA 算法的簇头数目 N_g 为

$$N_g \approx \left\lceil \frac{w}{w_1} \right\rceil \frac{h}{Adv_{ESA}} \quad (13)$$

其中, Adv_{ESA} 表示 ESA 算法中网格的平均高度.

对于一些查询任务固定的传感器网络应用, 查询信息通常直接存储于每个传感器节点中, 因而可以认为 $S_q = 0$. 在这种情况下, ESA 算法优于 IWQE 算法, 节省的能量大小约等于 $(N_q - 1) S_a E$.

另外, 当 $S_q \neq 0$ 且 $S_a \gg S_q$ 时, 则 $(\alpha N_q - N_g - 1) S_q E$ 的值相比 $(N_q - 1) S_a E$ 可以忽略不记, ESA 算法优于 IWQE 算法, 节省的能量大小约等于 $(N_q - 1) S_a E$. 当 $S_q \neq 0$ 且 $S_a \gg S_q$ 不成立时, 由于 $N_q(N_g)$ 的值与路线宽度(网格宽度)、节点密度、节点通信半径、网络拓扑、查询区域内节点分布等因素有关, 难以求解, 因而在第 4 节通过实验对其进行分析和比较.

对于连续空间范围查询(例如每隔 1 秒钟获得查询区域 sw 的平均温度), 由于发送感知数据的能量消耗与发送查询消息消耗的能量相比可以忽略不记. IWQE 算法与 ESA 算法消耗的能量之差为

$$E_{total}(IWQE) - E_{total}(ESA) \approx K(N_q - 1) S_a E \quad (14)$$

其中, K 表示连续空间范围查询对应的快照空间范围查询个数. 因此, ESA 算法优于 IWQE 算法, 节省的能量大小为 $K(N_q - 1) S_a E$.

4 仿真评价

为了对算法的性能进行比较, 我们在文献[21]的仿真器上实现了 IWQE 算法和 ESA 算法. 为了描述方便, 记采用 GAA(GBA)算法的 ESA 算法为 ESA-GAA(ESA-GBA). 实验的硬件环境为 P4 (2.66GHz) CPU, 512MB 内存; 软件环境为 Ubuntu 操作系统、Eclipse 开发工具. 实验的参数选择如下:

根据文献[22],无线通信模块发送和接收 1 字节数据的能量消耗公式为 $E_t = \alpha + \gamma \times l^e$, $E_r = \beta$. 采用文献[23]的参数: $\gamma = 10 \text{ pJ}/(\text{bit} \cdot \text{m}^2)$, $\alpha = 45 \text{ nJ}/\text{bit}$, $\beta = 135 \text{ nJ}/\text{bit}$, $e = 2$. 其它参数如表 1 所示.

表 1 实验参数

参数名	参数值
网络覆盖区域	100 m × 100 m
网络节点分布	随机均匀分布
节点通信半径	10 m
节点数	800
感知数据消息大小	40 bytes
查询消息大小	20 bytes
查询区域占网络覆盖区域的百分比	60%
网格宽度占通信半径的比率	$\sqrt{3}/2$
IWQE 算法路线宽度占通信半径的比率	$\sqrt{3}/2$

4.1 比较 ESA-GAA 和 ESA-GBA 的能量消耗

表 2 和表 3 分别显示了节点数目为 320 和 800 时,查询区域大小对 ESA-GAA 和 ESA-GBA 算法能量消耗的影响. 可见,当节点密度较小时,ESA-GAA 和 ESA-GBA 的能量消耗近似相等;当节点密度较大且查询区域较小时,在绝大多数情况下,ESA-GBA 略优于 ESA-GAA;当节点密度较大且查询区域较大时,在绝大多数情况下,ESA-GAA 略优于 ESA-GBA. 由于在各种实验参数条件下,ESA-GAA 与 ESA-GBA 能量消耗相差较小,在后面几组实验中,为了描述方便,仅给出 ESA-GBA 的实验结果.

表 2 节点数目为 320 时 ESA 的能量消耗

查询区域占网络覆盖区域的百分比	ESA-GAA 消耗的能量/mJ	ESA-GBA 消耗的能量/mJ
0.1	249.62	249.62
0.2	502.57	502.76
0.3	715.07	715.07
0.4	913.54	913.08
0.5	1142.05	1142.05
0.6	1264.11	1264.11
0.7	1479.47	1479.47
0.8	1696.90	1696.90
0.9	1907.29	1907.29

表 3 节点数目为 800 时 ESA 的能量消耗

查询区域占网络覆盖区域的百分比	ESA-GAA 消耗的能量/mJ	ESA-GBA 消耗的能量/mJ
0.1	1017.09	1017.16
0.2	2053.91	2052.61
0.3	3042.84	3042.71
0.4	4008.59	4006.32
0.5	5092.29	5092.71
0.6	5995.58	5995.84
0.7	6952.67	6952.76
0.8	7901.26	7900.72
0.9	8566.14	8569.39

4.2 比较 ESA 簇头节点数目和 IWQE 查询节点数目

由式(11)可知, $E_{\text{total}}(\text{IWQE})$ 与 $E_{\text{total}}(\text{ESA})$ 之差主要由 N_q 和 N_g 决定,本组实验分析 N_q 和 N_g 与节点密度、查询区域大小、网格大小(路线宽度)之间的关系.

图 11 显示了节点密度对 N_q 和 N_g 的影响. 可见, $N_g > N_q$, 且随着节点密度的增大, N_g 和 N_q 均变小, N_g 降低的幅度小于 N_q . 这是因为: 为了减少查询处理过程发送的数据消息数目, ESA 要求网格的大小和簇头节点的选择满足 ESA 约束条件而 IWQE 算法没有该约束. 因而 $Adv_{\text{ESA}} < Adv_{\text{IWQE}}$, 且当节点密度增大时, Adv_{ESA} 的增幅小于 Adv_{IWQE} .

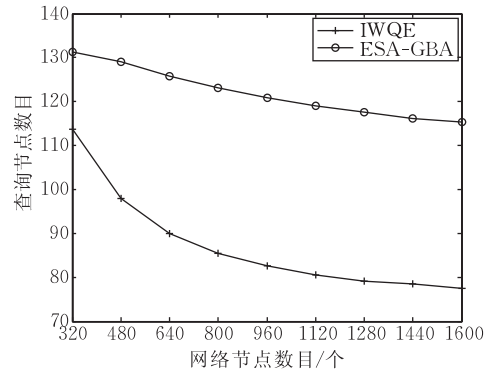
图 11 节点密度对 N_q 和 N_g 的影响

图 12 显示了查询区域大小对 N_q 和 N_g 的影响, 可见,当查询区域较小时, N_q 和 N_g 相差不大. 随着查询区域的增大,根据式(12)和(13)可知,可以近似认为 N_g 和 N_q 与查询区域面积成正比,因而 N_g 和 N_q 均变大.

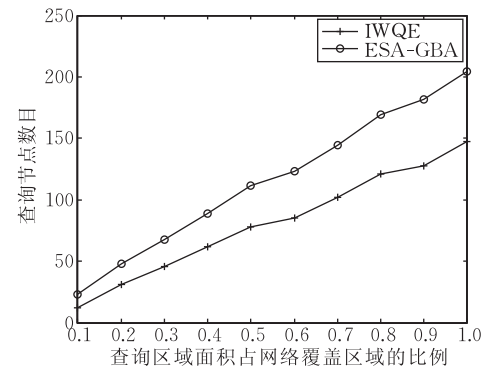
图 12 查询区域大小对 N_q 和 N_g 的影响

图 13 显示了网格宽度对 N_q 和 N_g 的影响. 实验中 IWQE 算法的路线宽度与 ESA 算法中的网格宽度相等. 可见,随着路线宽度的增大, Adv_{IWQE} 基本保持不变,根据式(12)可知, N_q 近似与路线宽度成反比,因而 N_q 逐渐变小. 对于 ESA 算法,当网格宽度较小时,查询路线较长,使得 N_q 较大;当网格宽度较

大时,为了满足 ESA 约束条件, Adv_{ESA} 较小,使得 N_q 较大. 因此,随着网格宽度的增大, N_q 呈现先减小后增大的趋势.

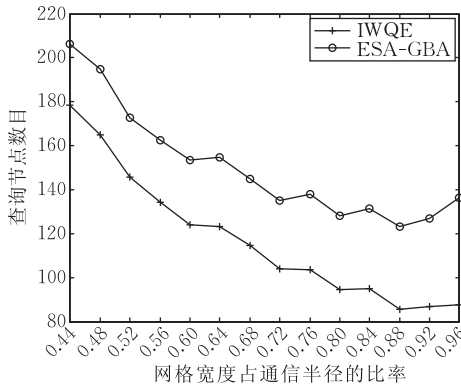


图 13 网格(路线)宽度对 $N_g(N_q)$ 的影响

4.3 比较 ESA 算法和 IWQE 算法的能量消耗

图 14 和图 15 分别显示了查询区域大小和节点密度对算法能量消耗的影响. 可见,当查询区域或节点密度较小时, N_q 和 N_g 相差不大且需要收集的感知数据数目较小,因而两种算法的能耗相差不大. 随着节点密度和查询区域的增大,两种算法因分发查询和收集感知数据消耗的能量均增大,因而总能量消耗均增大. 随着查询区域(节点密度)的增大, N_q 也随之增大,根据式(11)可知,相比 IWQE 算法,

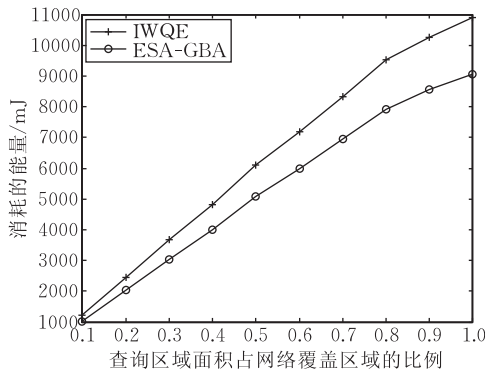


图 14 查询区域大小对能量消耗的影响

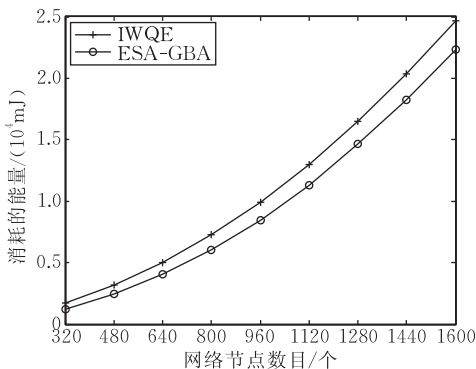


图 15 节点密度对能量消耗的影响

ESA 算法少发送了 $(N_q - 1)$ 个数据消息包,大大降低了算法的能量消耗. 当网络中的节点数目取 320~1600 之间时,ESA 算法减少的能量占 IWQE 算法总能耗的比例为 $[9.57\%, 27.57\%]$.

图 16 显示了网格宽度对能量消耗的影响. 可见,随着路线宽度的增大, N_q 变小,根据式(9)可知, IWQE 算法消耗的能量逐渐减小. 而对于 ESA 算法, N_q 呈现先减小后增大的趋势,根据式(3)可知,其能量消耗也呈现先减小后增大的趋势.

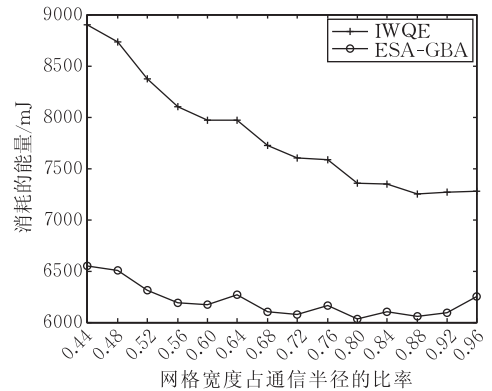


图 16 网格(路线)宽度对能量消耗的影响

4.4 比较 ESA 与 IWQE 的查询成功率和查询结果质量

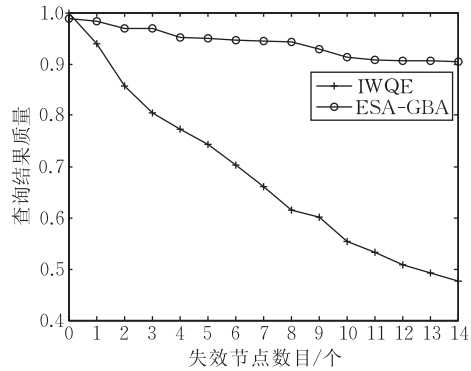
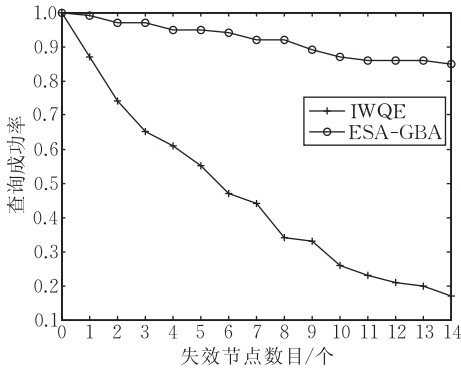
本组实验比较 IWQE 和 ESA 算法在不同失效节点数目、查询区域大小、节点密度和网格宽度条件下的查询成功率和查询结果质量. 查询成功率定义为 $sr = se/te$, 其中, te 表示算法运行的总次数, se 表示算法成功返回查询结果至基站的次数. 与文献[1]相同, 查询结果质量定义为 $rq = cn/tn$, 其中, tn 表示在查询区域内的节点数目, cn 表示在查询区域内且被算法遍历到的节点数目. 采用表 1 中的实验参数, 失效节点数目设置为 4, 即节点失效的比例为 0.5%.

图 17 显示了失效节点数目对查询成功率和查询结果质量的影响. 由于在 IWQE 算法中, 任意一个查询节点失效会导致查询结果全部丢失且无法返回查询结果, 因而 IWQE 算法的查询成功率和查询结果质量均随着失效节点数目的增大急剧下降. 而 ESA 算法在簇头节点失效时从冗余节点恢复执行, 降低了查询处理过程中断的概率, 因而其查询成功率和查询结果质量随着失效节点数目的增大较缓慢地下降.

图 18 显示了查询区域大小对查询成功率和查询结果质量的影响. 可见, 两种算法的查询成功率和查询结果质量随着查询区域的增大均下降, 且 IWQE

算法急剧下降,而 ESA 算法较缓慢的下降. 这是因为:在失效节点数目固定的情况下,随着查询区域的增大,算法执行过程中遇到失效节点的概率增大,导致两种算法的查询成功率和查询结果质量均下降.

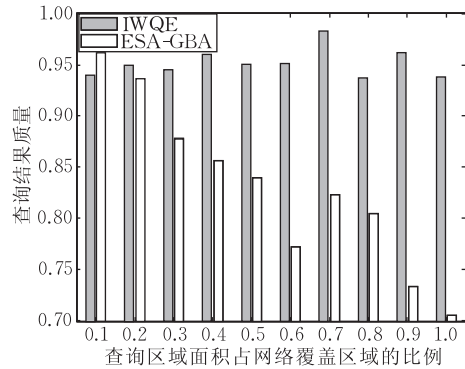
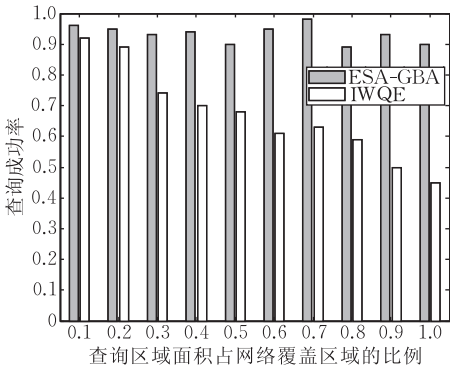
在 IWQE 算法中,任意一个查询节点失效则查询处理过程中止,而 ESA 算法只有在簇头节点及其所有备份节点同时失效的情况下才会中断,提高了查询处理过程的鲁棒性.



(a) 网络失效节点数目对查询成功率的影响

(b) 网络失效节点数目对查询结果质量的影响

图 17 失效节点数目对查询成功率和查询结果质量的影响



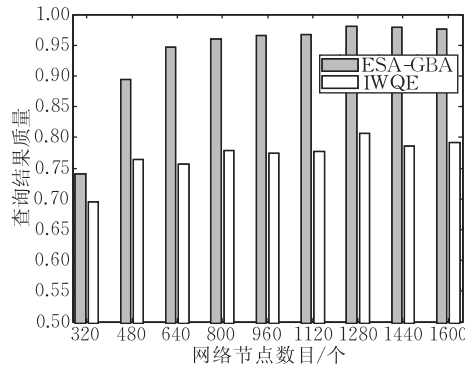
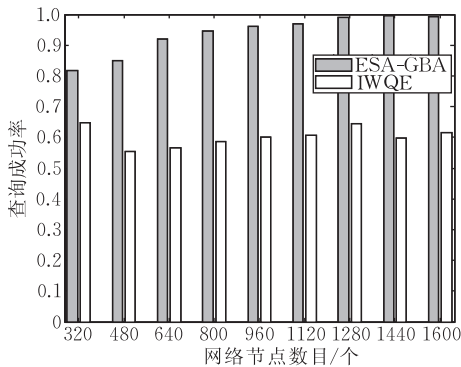
(a) 查询区域大小对查询成功率的影响

(b) 查询区域大小对查询结果质量的影响

图 18 查询区域对查询成功率和查询结果质量的影响

图 19 显示了网络节点密度对查询成功率和查询结果质量的影响. 节点数目分别取 $160 \times 2, 160 \times 3, \dots, 160 \times 10$, 失效节点数目分别取 $2, 3, \dots, 10$. 可见,在失效节点比例固定的情况下,随着网络节点密度的增大, IWQE 算法的查询成功率和查询结果质量基本不变,而 ESA 的查询成功率和查询结果质

量有显著的提高. 尤其,当节点密度较大时,节点失效对 ESA 的影响可以忽略不计. 其原因是:当网络节点密度变大时,ESA 算法簇头节点的备份节点数目也随之增大,有效地降低了查询处理中断的概率.



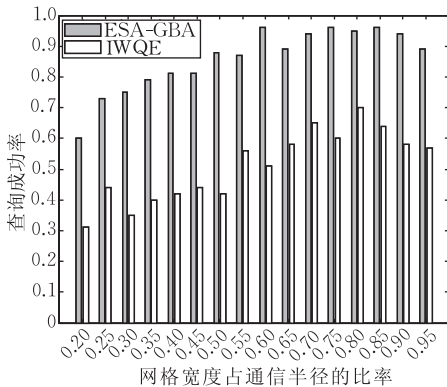
(a) 网络节点密度对查询成功率的影响

(b) 网络节点密度对查询结果质量的影响

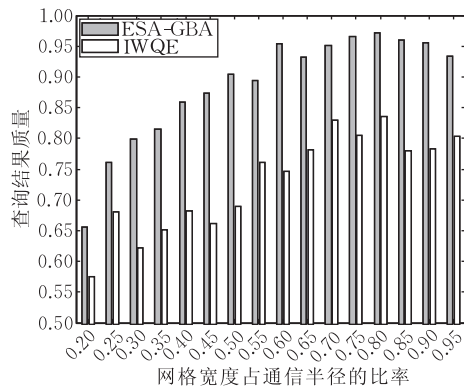
图 19 网络节点密度对查询成功率和查询结果质量的影响

图 20 显示了网格宽度对查询成功率和查询结果质量的影响。在实验中, IWQE 算法的路线宽度取值与 ESA 算法的网格宽度相同。可见, 随着网格宽度的增大, IWQE 和 ESA 算法的查询成功率和查询结果质量均呈现先增大后减少的趋势。其原因是: 当 ESA 算法的网格宽度 (IWQE 算法的路线宽度) 较小时, 两种算法的查询路线较长, 使得 N_g 和 N_q 较大。对于 IWQE 算法, N_q 越大, 则查询处理过程因查询节点失效的概率越大; 对于 ESA 算法, 网格的平

均面积与 N_g 成反比, 使得簇头节点的备份节点数目减少, 因而 IWQE 和 ESA 算法的查询成功率较低且查询结果质量较差; 当 ESA 算法的网格宽度 (IWQE 算法的路线宽度) 较大时, 对于 ESA 算法, 为了满足 ESA 约束条件, Adv_{ESA} 较小, 使得 N_g 较大; 对于 IWQE 算法, 查询区域内不在查询节点通信半径内的数据节点数目随着路线宽度变大而增大, 使得查询成功率较低且查询结果质量较差。



(a) 网格宽度对查询成功率的影响



(b) 网格宽度对查询结果质量的影响

图 20 网格宽度对查询成功率和查询结果质量的影响

5 总结与展望

现有传感器网络空间范围查询处理算法能量消耗大且查询处理过程容易因节点失效而中断, 无法获得查询结果。本文提出了一种综合考虑能量消耗和节点失效的算法 ESA, 它仅要求查询区域内的每个节点发送一次数据消息, 降低了能量消耗。为了减小查询处理因节点失效而中断的概率, 我们设计了一种利用节点冗余恢复查询处理过程的算法, 实验结果表明该算法有效地保证了 ESA 算法查询处理过程的查询成功率和查询结果质量, 且随着节点密度的增大, ESA 算法的查询成功率和查询结果质量显著提高。尤其当节点密度较大时, 节点失效对 ESA 算法查询处理的影响可以忽略。因此, 在绝大多数情况下, 本文提出的算法优于 IWQE 算法。

文献[18-20]采用了 IWQE 类似的思想处理 K 近邻查询, 本文指出的问题在这些算法中依然存在, 将 ESA 算法中的思想应用于 K 近邻查询算法是我们下一步的工作。

致 谢 感谢加拿大 Alberta 大学 Alexandru Coman 以及审稿专家, 他们提出了宝贵的意见!

参 考 文 献

- [1] Xu Ying-Qi, Lee Wang-Chien, Xu Jian-Liang et al. Processing window queries in wireless sensor networks//Proceedings of the 22nd International Conference on Data Engineering. Atlanta, USA, 2006: 70-80
- [2] Madden S, Franklin M, Hellerstein J et al. TAG: A tiny aggregation service for ad hoc sensor networks//Proceedings of the 5th Symposia on Operating Systems Design and Implementation. Boston, USA, 2002: 131-146
- [3] Goldin D, Song M, Kutlu A et al. Georouting and delta-gathering: Efficient data propagation techniques for geosensor networks//Proceedings of the 1st GeoSensor Networks Workshop. Portland, USA, 2003: 1-23
- [4] Murat D, Hakan F. Peer-to-Peer spatial queries in sensor networks//Proceedings of the 3rd International Conference on Peer-to-Peer Computing. Linköping, Sweden, 2003: 32-39
- [5] Amir S, Vana K, Dimitrios G. Spatial queries in sensor networks//Proceedings of the 13th ACM International Workshop on Geographic Information Systems. Bremen, Germany, 2005: 61-70
- [6] Coman A, Nascimento M A, Sander J. A framework for spatio-temporal query processing over wireless sensor networks//Proceedings of the 1st International Workshop on Data Management for Sensor Networks in Conjunction with Very large Database 2004. Toronto, Canada, 2004: 104-110

- [7] Jain N, Yalagandula P, Dahlin M et al. Self-tuning, bandwidth-aware monitoring for dynamic data streams//Proceedings of the 22nd International Conference on Data Engineering. Shanghai, China, 2009; 114-125
- [8] Jain N, Yalagandula P, Dahlin M et al. STAR: Self-tuning aggregation for scalable monitoring//Proceedings of the 33rd International Conference on Very Large Data Bases. Vienna, Austria, 2007; 962-973
- [9] Tang Xue-Yan, Xu Jian-Liang. Adaptive data collection strategies for lifetime-constrained wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(6): 721-734
- [10] Tang Xue-Yan, Xu Jian-Liang. Optimizing lifetime for continuous data aggregation with precision guarantees in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 2008, 16(4): 904-917
- [11] Wu Min-Ji, Xu Jian-Liang, Tang Xue-Yan et al. Top- k Monitoring in wireless sensor networks. *IEEE Transactions Knowledge Data Engineering*, 2007, 19(7): 962-976
- [12] Chen Bai-Chen, Liang Wei-Fa, Zhou Rui et al. Energy-efficient top- k query processing in wireless sensor networks//Proceedings of the 19th ACM Conference on Information and Knowledge Management. Toronto, Canada, 2010; 329-338
- [13] Guttman A. R-Trees: A dynamic index structure for spatial searching//Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data. Boston, USA, 1984; 47-57
- [14] Karp B, Kung H T. GPSR: Greedy perimeter stateless routing for wireless networks//Proceedings of the 6th Annual International Conference on Mobile Computing and Networking. Boston, USA, 2000; 243-254
- [15] Cheng Si-Yao, Li Jian-Zhong. Sampling based (ϵ, δ)-approximate aggregation algorithm in sensor networks//Proceedings of the 29th IEEE International Conference on Distributed Computing Systems. Montreal, Canada, 2009; 273-280
- [16] Cheng Si-Yao, Li Jian-Zhong, Ren Qian-Qian et al. Bernoulli sampling based (ϵ, δ)-approximate aggregation in larger-scale sensor networks//Proceedings of the 29th IEEE International Conference on Computer Communications. San Diego, USA, 2010; 1181-1189
- [17] Sharifzadeh M, Shahabi C. Supporting spatial aggregation in sensor network databases//Proceedings of the 12th ACM International Workshop on Geographic Information Systems. Washington, USA, 2004; 166-175
- [18] Wu Shan-Hung, Chuang Kun-Ta, Chen Chung-Min et al. DIKNN: An itinerary-based KNN query processing algorithm for mobile sensor networks//Proceedings of the 23rd International Conference on Data Engineering. Istanbul, Turkey, 2007; 456-465
- [19] Wu Shan-Hung, Chuang Kun-Ta, Chen Chung-Min et al. Toward the optimal itinerary-based KNN query processing in mobile sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 2008, 20(12): 1655-1668
- [20] Fu Tao-Yang, Peng Wen-Chih, Lee Wang-Chien. Parallelizing itinerary-based KNN query processing in wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 2010, 22(5): 711-729
- [21] Coman A, Sander J, Nascimento M A. Adaptive processing of historical spatial range queries in peer-to-peer sensor networks. *Distributed and Parallel Databases*, 2007, 22(2-3): 133-163
- [22] Rappaport T. *Wireless Communications: Principles and Practice*. New Jersey: Prentice-Hall Press, 1996
- [23] Bhardwaj M. *Power-Aware Systems*. http://www.mtl.mit.edu/researchgroups/icsystems/pubs/theses/manishb_sm_2001.pdf, 2010-03-03



LIU Liang, born in 1985, Ph. D. candidate. His research interests focus on data management in distributed environment.

QIN Xiao-Lin, born in 1953, professor, Ph. D. supervi-

sor. His research interests include security database, temporal-spatial database, data management and security in distributed environment

ZHENG Gui-Neng, born in 1988, M. S candidate. His research interests focus on data management in wireless sensor networks.

LI Bo-Han, born in 1979, Ph. D. . His research interests focus on data management in wireless sensor networks.

Background

WSNs (Wireless Sensor networks) have a wide range of applications in military defense, medical services, traffic control and etc. WSN is a kind of distributed embedded system. Sensor nodes have very limited computing power, storage capacity and energy. Therefore, the development of sensor net-

work applications is extremely difficult. Unlike other networks, wireless sensor networks are data-centric. The main purpose of using WSNs by the users is to collect the events or data generated. WSN data management system provides users with a simple query interface which shields the com-

plexity of query processing. It greatly simplifies the development of sensor network applications. It has become an important middleware for WSNs. Query processing is the core technology of WSN data management system. Although extensive research has been conducted on query processing techniques in wireless sensor networks, these studies are based on many assumptions. For example, sensor nodes don't fail and network topology never changes. However, they are not valid in realistic deployment. Therefore, the state-of-the-art query processing algorithms have fairly high energy consumption, low query success rate and incorrect query result.

In many wireless sensor network applications, users often submit spatial window queries to obtain summary information about a local area in the monitored region such as the maximum temperature, average humidity and so on. The authors propose an energy-efficient spatial window query processing algorithm called ESA in this paper. ESA only requires each node within query region and sends sensory data

message once, which reduces the number of sensory data messages. They propose two grid dividing and cluster node selection algorithms according to the ESA's energy consumption formula to reduce the energy consumption of distributing the query messages, then, design a query processing recovery algorithm using node redundancy, which reduces the outage probability caused by node failures. Finally, the performance of ESA and IWQE is analyzed systematically. Analytical and experimental results show that ESA outperforms IWQE in most cases. This work is supported by the National Natural Science Foundation of China (60673127), the National High Technology Research and Development Program (863 Program) of China (2007AA01Z404), the Electronic Development Fund of the Ministry of Information Industry, the Jiangsu Province Science & Technology Pillar Program (BE2008135), the China Postdoctoral Science Foundation (20100481133) and the Jiangsu Province Postdoctoral Science Foundation (1001005B).